Netzwerksicherheit: Traces, Logs und die Rolle der Anonymisierung

Informatiktag 2025

HTW Berlin, 17.06.2025
Prof. Dr. Thomas Scheffler (thomas.scheffler@htw-berlin.de)

Mein Hintergrund

> 25 Jahre Erfahrung im Bereich Netzwerke und Netzwerksicherheit

- Netzwerkprovider
- Hochschule in unterschiedlichen Rollen
  - Lehrender
  - Forschender
  - Informationssicherheitsbeauftragter

Netzwerksicherheit: Traces, Logs und die Rolle der Anonymisierung

Einführung

- **Thema**: Bedeutung von Trace- und Logdateien in der Netzwerksicherheit
- **Ziel**: Verständnis für die Nutzung und Herausforderungen im Umgang mit Netzwerkdaten
- **Kernthemen**:
  - Datenschutzprobleme und Rolle der Anonymisierung

Bedeutung von Traces und Logs

Traces und Logs sind wichtige Datenquellen zur Erkennung von Anomalien

- **Funktion:** Überwachung des Datenverkehrs zur Erkennung und Abwehr böswilliger Aktivitäten
- **Ziel**: Schutz von Netzwerken durch detaillierte Analyse des Datenverkehrs, Entwicklung neuer Werkzeuge zur Angriffserkennung, Erkennung von Trends

Beispielanwendung: Intrusion-Detektionssysteme (IDS)

- **Regelbasierte Systeme**: Erkennen definierter bösartiger Muster
- **Heuristische Verfahren**: Näherungslösungen, Anwendung von Erfahrungswerten
- **KI in IDS**:
  - **Motivation**: Automatische Mustererkennung
  - **Vorteile**: Skalierbarkeit und präzisere Erkennung
  - **Forschung**: Entwicklung intelligenter Algorithmen

Rechtliche Rahmenbedingungen in Deutschland

- **Datenschutzgesetze**: EU-DSGVO und Bundesdatenschutzgesetz
- **Löschpflicht**: Logfiles müssen spätestens nach 90 Tagen gelöscht werden
- **Zweckbindung**: Nutzung von Daten nur für definierte, rechtmäßige Zwecke (z.B. Aufrechthaltung des Betriebs)
- **Einwilligung**: Erforderlichkeit der Zustimmung des Betroffenen oder gesetzliche Erlaubnis

Öffentliche Netzwerk-Traces

- **Bedeutung:**
  - Öffentliche Netzwerk-Traces sind entscheidend für die Entwicklung und Evaluierung neuer Sicherheitslösungen unter realen Bedingungen.
  - Sie ermöglichen die Analyse von Bedrohungsszenarien und das Training von Machine-Learning-Modellen.
- **Herausforderungen:**
  - *Relevanz der Daten*: Öffentliche Traces sind teilweise veraltet und reflektieren nicht alle aktuellen Bedrohungen und Angriffstechniken
  - *Datenschutz*: Strenge Datenschutzvorgaben erschweren die Bereitstellung aktueller und realer Netzwerkdaten (extern aber auch intern)

Beispiele für öffentlich zugängliche Traces:

- MAWI Working Group Traffic Archive: Beinhaltet Langzeit-Traces von japanischen Backbone-Netzen
- CAIDA (Center for Applied Internet Data Analysis): Bietet verschiedene Datensätze für Internet-Forschung, inklusive Anonymisierungsrichtlinien
- DARPA Intrusion Detection Data Sets: Beinhalten simulierte Cyberangriffe für Evaluierungszwecke (nicht mehr aktiv bereitgestellt)
- KDD Cup 1999 Data
- PREDICT (Protected Repository for the Defense of Infrastructure Against Cyber Threats): Bereitstellt Zugang zu Netzwerkdatensätzen für die Cyber-Sicherheitsforschung (nicht mehr aktiv bereitgestellt)

Nutzung dieser Traces:

- Analyse von Netzwerkverhalten und Protokollmustern
- Entwicklung und Bewertung von IDS und anderen Sicherheitsmechanismen
- Überprüfung und Validierung von Forschungsansätzen im Netzwerksicherheitsbereich
- **Limitierungen:**
  - **Simulation vs. Realität:** Einige Traces basieren auf simulierten Umgebungen, was die Übertragbarkeit auf reale Systeme einschränken kann
  - **Relevanz:** Mangelnde Aktualität und geografische Begrenzung können die Relevanz einschränken

Die Rolle der Anonymisierung

- **Anonymisierungstechniken**:
  - Herstellung der Datenschutzkonformität durch technische Maßnahmen
  - Daten können anschließend sicher aufbewahrt und ggf. weitergeben werden
- **Werkzeuge**:
  - **Anonymisierung von IP-Adressen**: Techniken zur Verminderung/Eliminierung personenbezogener Bezüge

Anonymisierung von IP-Adressen

1. *Black-marker* - Teile der IP-Adressen werden 'gelöscht'. Erlaubt keine Korrelation von Ereignissen, Adress-Information gehen verloren.
2. *Random permutation* - Zufälliges 1:1 Mapping, Information über Netzwerkstruktur geht verloren.
3. *Prefix-Preserving* - Netzwerkstrukturen bleiben im Verfahren erhalten, da identische Bitkombinationen der 'high-order-bits' auf gleiche Ausgabe-Bitmuster mappen.

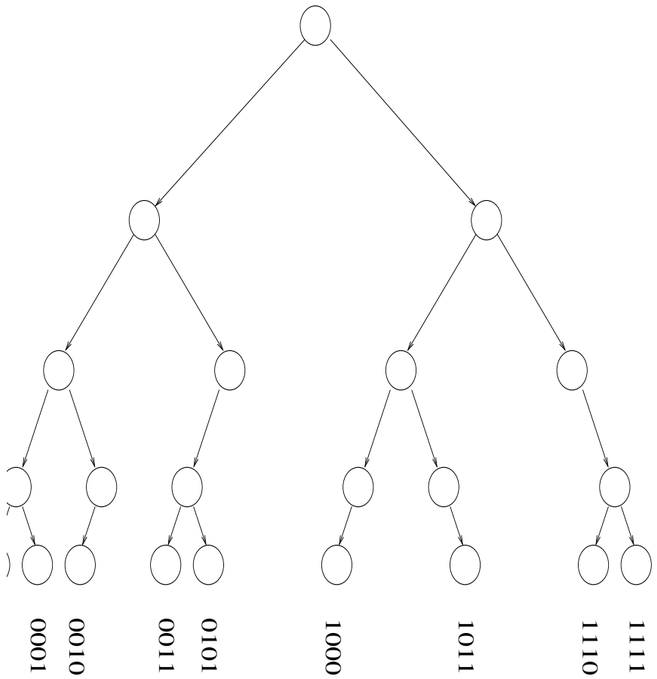
Einführung in CryptoPAn

- **Cryptography-based Prefix-preserving Anonymization (CryptoPAn)**
  - **Tool zur Anonymisierung von IP-Adressen**: Erhält statistische Eigenschaften des Netzwerkverkehrs
- **Vorteile von CryptoPAn**:
  - **Präfix-Preserving Anonymisation**: Beibehaltung der Netzwerkstruktur
  - **Konsistenz**: Gleiche Adressen werden immer gleich anonymisiert (Algorithmus benutzt einen symmetrischen Schlüssel)
- **Anwendungsbereiche**:
  - Nützlich für die Forschung und Netzwerkanalyse
  - Unterstützung bei der Entwicklung und Evaluierung neuer Sicherheitssysteme

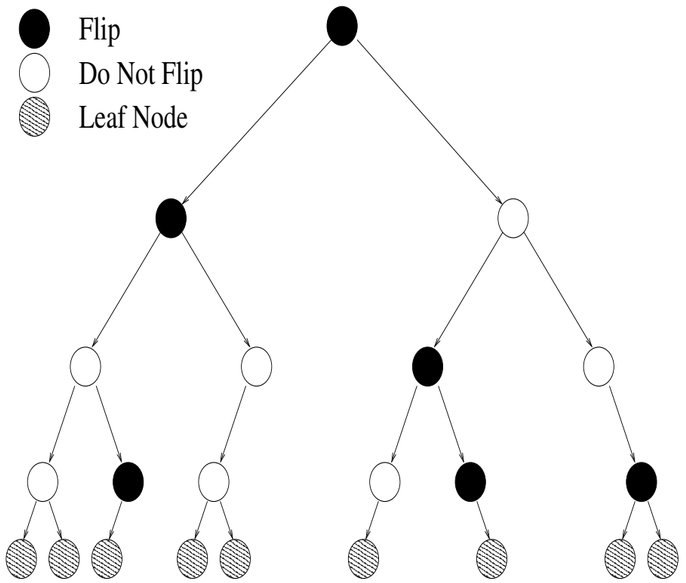
Der Algorithmus von CryptoPAn

- **Präfix-Erhaltung**: Der Algorithmus anonymisiert IP-Adressen so, dass die Präfix-Struktur der Adresse erhalten bleibt. Benachbarte Adressbereiche sind auch nach der Transformation benachbart.
- **Kryptografische Zeichenkette**: Verwendung einer Pseudorandom-Funktion (PRF) zur Generierung konsistenter und für Außenstehede nicht rückverfolgbarer Anonymisierungen.
- Wenn der geheime Schlüssel zur Anonymisierung von IP-Adressen bekannt ist, sind die anonymisierten Adressen potenziell reversibel.

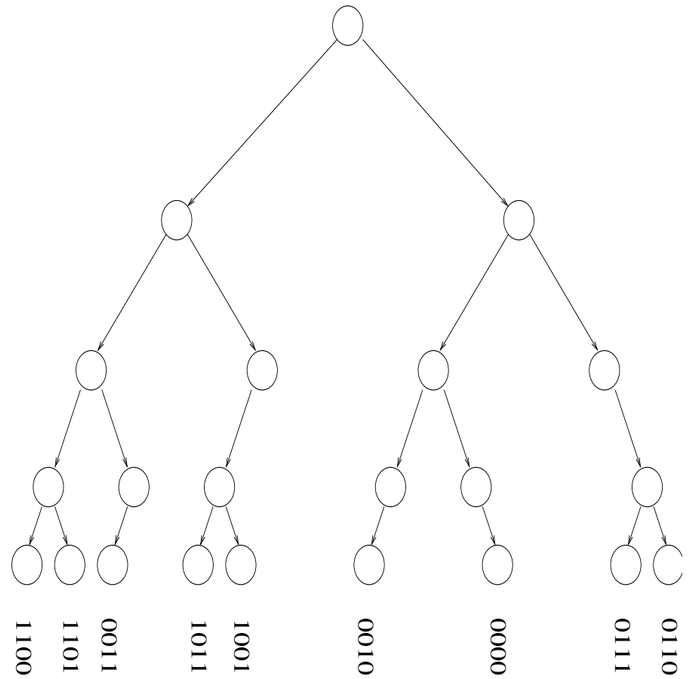
Der Algorithmus von CryptoPAn



b) original address tree



(c) anonymization function



(d) anonymized address tre

Limitierungen von CryptoPAn

- **Rückverfolgbarkeit**: Trotz Anonymisierung bleibt ein gewisses Maß an Rückverfolgbarkeit möglich durch externe Korrelation von Daten.
- **Spezialadressen**: Es werden auch Private IPs, Multicast und Spezialadressen (Localhost) anonymisiert. Diese Adressen beinhalten keine privaten Daten, sind aber für die Angriffsanalyse und Erkennung sehr hilfreich (Bogons, Adress-Spoofing).
- **Randomness**: Durch die Muster-erhaltende Art der Anonymisierung ist keine perfekte Randomisierung des Adressraums möglich.
- **Geheimhaltung**: Der Schlüssel für die Anonymisierung ist geheim zu halten und kann praktisch nur in größeren Zeiträumen (z.B. 1 Jahr) gewechselt werden.

Herausforderungen und Forschung

- **Tech & Recht**: Schwierigkeiten in der praktischen Umsetzung von Anonymisierung
- **Forschungsbedarf**:
  - **Verbesserung von Anonymisierungstechniken**: Effizienz und Nutzerfreundlichkeit
  - **Klärung rechtlicher Rahmenbedingungen**: Konformität mit Datenschutzgesetzen sicherstellen
  - *Best practices* für die Anwendung von Anonymisierungsmethoden wie CryptoPAn, u.a.
  - Regulatorische Klarheit und technologischer Fortschritt als Schlüssel für mehr IT-Sicherheit 'Made in Germany'

Diskussion

- Offene Fragen zur technischen Eignung und der rechtlichen Umsetzung
- Zukunftsaussichten für die IT-Sec Forschung in Deutschland und darüber hinaus:
  - Datenschutz ist ein relevantes Merkmal europäischer IT
  - Derzeit aber oft eine juristische Debatte - Anreize für neue technische Verfahren und Forschung fehlen mitunter

Quellen

- https://conferences.sigcomm.org/imc/2001/imw2001-papers/69.pdf
- https://faculty.cc.gatech.edu/~mbailey/publications/catch09*anonymizing*final.pdf
- https://ieeexplore.ieee.org/document/1181415
- https://en.wikipedia.org/wiki/Crypto-PAn

CryptoPAN Algorithmus

$$f_i(a_1 a_2 \dots a_i) := L(R(P(a_1 a_2 \dots a_i); k))$$

- L - gibt das 'least significant bit' zurück.
- R - Pseudozufallsfunktion (hier AES).
- P - Padding-Funktion, erzeugt Blockgrößen passend zu R.

$I_i = x_{[0,i)}\ \mathbf{pad}_{[i,128)}$, das Padding bringt die Eingabe auf die erforderliche Größe.

Bitweiser Verschlüsselungsprozess:
- Für jedes Bit x_i der ursprünglichen IPv4-Adresse x:

1. **Erstelle den Eingabeblocks I_i:**
   — Extrahiere die Adressbits von x von Anfang an bis, aber ohne, das aktuelle Bit x_i, Teilpräfix der Adresse: $x[0, i)$.
   — Fülle dieses Präfix auf, um einen 128-Bit-Block zu bilden. Das Padding gewährleistet eine einheitliche Eingangsgröße für die Verschlüsselungsfunktion. Das Padding füllt den Bereich von Bit i bis Bit 128 aus. $I_i = x_{[0,i)}\ \mathbf{pad}_{[i,128)}$
2. **Verschlüsseln des Eingabeblocks:**
   — Verwende einen symmetrischen Verschlüsselungsalgorithmus, um den 128-Bit-Eingabeblock I_i zu verschlüsseln, was einen 128-Bit-Ausgabeblock O_i ergibt.
   — Verwende einen kryptografischen Algorithmus wie AES, der einen geheimen Schlüssel k verwendet.
3. **XOR-Operation:**
   — Führe eine bitweise XOR-Operation zwischen dem i-ten Bit des Ausgabeblocks O_i und dem ursprünglichen Bit x_i durch: $x_i \oplus O_{i,i}$.
   — Hänge das resultierende Bit an den Ausgabebitstring an.
4. **Zusammensetzen der Ausgabe:**
   — Wiederhole diesen Prozess für alle 32 Bit der IPv4-Adresse, der Ausgabebitstring $E_k\ (x)$ entspricht der anonymisierten Adresse.

160.1.1.1 -> '164.1.1.130'
161.1.1.1 -> '165.1.14.0'
162.1.1.1 -> '167.193.13.3'
163.1.1.1 -> '166.206.242.0'
164.1.1.1 -> '163.49.14.253'
165.1.1.1 -> '162.254.241.255'
166.1.1.1 -> '160.254.254.243'
167.1.1.1 -> '161.62.242.241'
168.1.1.1 -> '168.62.254.113'
169.1.1.1 -> '169.62.254.253'
170.1.1.1 -> '171.241.2.3'
171.1.1.1 -> '170.254.241.114'
172.1.1.1 -> '172.206.254.115'
173.1.1.1 -> '173.1.14.113'
174.1.1.1 -> '175.254.253.131'
175.1.1.1 -> '174.241.2.129'
176.1.1.1 -> '183.254.253.142'
177.1.1.1 -> '182.254.241.130'
178.1.1.1 -> '180.49.1.2'
179.1.1.1 -> '181.62.242.241'
180.1.1.1 -> '179.206.254.255'
181.1.1.1 -> '178.206.253.240'
182.1.1.1 -> '176.241.1.12'
183.1.1.1 -> '177.14.242.143'
184.1.1.1 -> '184.14.241.128'
185.1.1.1 -> '185.49.13.1'
186.1.1.1 -> '186.14.253.131'
187.1.1.1 -> '187.49.14.140'
188.1.1.1 -> '188.1.1.241'
189.1.1.1 -> '189.14.241.142'
190.1.1.1 -> '190.49.13.2'